LSB embedding

- Nosič $x = (x_1, \dots, x_n) \in \mathbb{Z}^n$

  např. x_i intenzita jasu $i$tého pixelu
  x_i intenzita červené složky $i$tého pixelu
  index do palety v obrázku GIF
  kvantizovaný DCT koeficient

- Stegoobjekt $y = (y_1, \dots, y_n) \in \mathbb{Z}^n$

- Zpráva $z = (z_1, \dots, z_m) \in \{0,1\}^m$, kde $m \leq n$.

- Klíč $\pi \in S_n$

- Vkládání: $y_{\pi(i)} = x_{\pi(i)} - (x_{\pi(i)} \bmod 2) + z_i$ pro $i = 1, \dots, m$.
  $y_{\pi(i)} = x_{\pi(i)}$ pro $i = m+1, \dots, n$.

- Platí $|x_i - y_i| \leq 1 \ \forall i$ (t_j provádíme nejmenší možnou změnu).

LSB rovina nekomprimovaných obrázků vypadá vesměs náhodně kvůli šumu při snímání obrázku. Vložení náhodné zprávy by tedy zdánlivě nemělo být detekovatelné. Ve skutečnosti existují mezi sousedními pixely vztahy které jsou tímto vkládáním narušeny.

- Platí $x_i \in \{2k, 2k+1\} \iff y_i \in \{2k, 2k+1\}$ čili: $0 \leftrightarrow 1$, $2 \leftrightarrow 3$, $4 \leftrightarrow 5$, $-2 \leftrightarrow -1$

- Tato vlastnost se projeví v histogramu.

  Definujeme $h_0(v) = \#\{i \mid x_i = v\}$ histogram nosiče
  $h(v) = \#\{i \mid y_i = v\}$ histogram stegoobjektu

  Platí $h(2k) + h(2k+1) = h_0(2k) + h_0(2k+1)$.

- Označme $\alpha = m/n$ a předpokládejme, že vkládáme náhodnou zprávu, potom

  $P(y_i \neq x_i) = \alpha \cdot 1/2$
  ↑ pravděpodobnost, že daný pixel obsahuje bit zprávy
  ← pravděpodobnost, že vložený bit se liší od původního LSB.

$$P(y_i = x_i) = 1 - \alpha/2$$

$$E[h(2k)] = \left(1-\frac{\alpha}{2}\right) h_0(2k) + \frac{\alpha}{2} h_0(2k+1)$$
$$E[h(2k+1)] = \frac{\alpha}{2} h_0(2k) + \left(1-\frac{\alpha}{2}\right) h_0(2k+1) \qquad (*)$$

Histogramový útok na úplný LSB embedding.

Jestliže $\alpha = 1$ pak pro všechna k

$$E[h(2k)] = E[h(2k+1)] = \frac{h_0(2k)+h_0(2k+1)}{2} = \frac{h(2k)+h(2k+1)}{2} =: \bar{h}(2k)$$

Čili $h(2k) \approx \bar{h}(2k)$.

Máme-li rozhodnout, zda $\alpha = 1$ nebo $\alpha \ll 1$ testujeme hypotézu, že h má rozdělení $\bar{h}$:

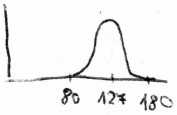
H_0: $h \sim \bar{h}$ (nulová hypotéza) $\alpha = 1$
H_1: $h \not\sim \bar{h}$ (alternativní hypotéza) $\alpha \ll 1$

- Použijeme Pearsonův χ^2 test. Spočítáme statistiku S:

$$S = \sum_{k=0}^{d-1} \frac{\left(h(2k) - \bar{h}(2k)\right)^2}{\bar{h}(2k)}$$

např. $d=128$
já myslím, že $d-1=128$

- Je zřejmé, že velká hodnota S indikuje $h \not\sim \bar{h}$ čili $\alpha < 1$, ale co znamená „velká hodnota"?
- Jestliže $\bar{h}(2k) > 4$ $\forall k$ a platí $h \sim \bar{h}$, pak ^{velicina} S má přibližně rozdělení χ^2_{d-1} s $d-1$ stupni volnosti.



- Hustota pravděpodobnosti rozdělení χ^2_{d-1} je $f_{\chi^2_{d-1}}(x) = \frac{e^{-x/2} \times \frac{d-1}{2}-1}{2^{\frac{d-1}{2}} \Gamma\left(\frac{d-1}{2}\right)}$.

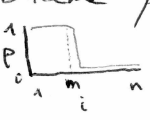
Čili $p = \int_S^\infty f_{\chi^2_{d-1}}(x)\,dx$ je pravděpodobnost, že S dosáhne takto vysoké (nebo vyšší) hodnoty za předpokladu $h \sim \bar{h}$.

Je-li $p \leq 0.05$ (např.) pak nulovou hypotézu zamítneme a říkáme, že „zamítáme H_0 (tj. $\alpha = 1$) na hladině významnosti 0.05%".

Čili: $(S \geq 154.3) \Leftrightarrow p \leq 0.05 \Rightarrow \alpha < 1$
$(S < 154.3) \Leftrightarrow p > 0.05 \Rightarrow \alpha = 1$

~~(pravděpodobnost, že zamítneme $\alpha = 1$ ačkoli~~ pravděpodobnost, že chybně zamítneme hypotézu $\alpha = 1$ je 5%.

- Známe-li π, můžeme odhadnout hodnotu α. Budeme provádět statistiku na $(x_{\pi(1)}, \ldots, x_{\pi(i)})$ pro $i = 1 \ldots n$ a počítat hodnotu p:



Kvantitativní útok na Jsteg

kvantitativní = určíme α

LSB 3

- Jsteg vkládá zprávu do LSB DCT koeficientů obrázku JPEG, s výjimkou koeficientů s hodnotami 0 a 1.
  Vkládání do 0 se totiž projeví vizuálně.

- Víme, že DCT koeficienty v JPEGu mají symetrický histogram
  $h_0[i] \approx h_0[-i] \quad i = 1, 2 \ldots$
  proto platí

$$\sum_{k>0} h_0(2k) - \sum_{k<0} h_0(2k) - \sum_{k\geq 0} h_0(2k+1) + \sum_{k<0} h_0(2k+1) \approx 0 \qquad (F)$$

(1,3,5,... ; -1,-3,-5,...)

- Soustavu (lineárních rovnic)(x) na str. LSB 2 můžeme přepsat:

$$h_0(2k) = a\,E[h(2k)] - b\,E[h(2k+1)]$$
$$h_0(2k+1) = -b\,E[h(2k)] + a\,E[h(2k+1)]$$

pro $k \neq 0$! protože do $\{0,1\}$ nevkládá.

kde $a = \frac{1-\alpha/2}{1-\alpha}$, $b = \frac{\alpha/2}{1-\alpha}$.

- Provedeme aproximaci $h(i) \approx E[h(i)]$ a dosadíme do (F).

$$\sum_{k>0} a\,h(2k) - b\,h(2k+1) - \sum_{k<0} a\,h(2k) - b\,h(2k+1)$$
$$- \sum_{k>0} -b\,h(2k) + a\,h(2k+1) + \sum_{k<0} -b\,h(2k) + a\,h(2k+1) \approx h_0(1) = h(1)$$

Pozor $k<0$ přesouváme na pravou stranu.

↑ do koef. 1 se nevkládalo

$$(a+b)\sum_{k>0} h(2k) - h(2k+1) + (a+b)\sum_{k<0} h(2k+1) - h(2k) \approx h(1)$$

$$a+b = \frac{1}{1-\alpha}$$

- Vyřešíme α:

$$\alpha \approx 1 - \frac{\sum_{k>0} h(2k) - h(2k+1) + \sum_{k<0} h(2k+1) - h(2k)}{h(1)}$$

α obvykle vychází s přesností ± 0.05

Sample pairs analysis (Analýza dvojic vzorků)

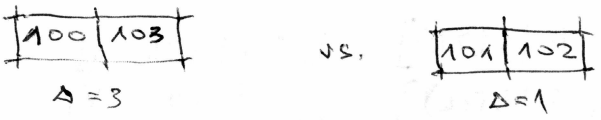
- Princip histogramového útoku: 

čekáme 

funguje pro $\alpha=1$ vs $\alpha\neq 1$

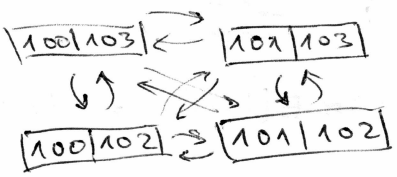
Pro obecné α bychom potřebovali vědět více o tvaru histogramu nosiče: Problém je v tom, že histogramy nosičů jsou příliš různorodé na to, abychom mohli vypreparovat nějakou obecnou vlastnost.

- Podíváme se na páry sousedních pixelů.
V přirozeném obrázku očekáváme, že sousední pixely budou mít blízké hodnoty. Například:



méně častý výskyt — častější výskyt

co se stane při LSB:



(Výskyty se vyrovnají.)

Jde v podstatě o to, že histogram párů sousedních pixelů má předvídatelný tvar. Přesněji jde o to, že v nosiči by výskyty těchto čtyř párů měly být nelineárně.

- Obecně: Máme čtveřice párů; jak vypadají jejich rozdíly?

	$(2i,2j)$	$(2i,2j+1)$	$(2i+1,2j)$	$(2i+1,2j+1)$	
$i<j$	d^{X}	$d+1^{Y_1}$	$d-1^{X}$	d^{Y_1}	$d=2j-2i$
$i>j$	d^{Y_1}	$d-1^{X}$	$d+1^{Y_1}$	d^{X}	$d=2i-2j$
$i=j$	0^{Z}	1^{Y_2}	1^{Y_2}	0^{Z}	

BTW: nulové vzdálenosti se objevují pouze v Z
jednotkové jedna v Y_2
jedna v X !!!

- Nechť P je multimnožina všech párů sousedních pixelů.

$$X=\{(r,s)\in P \mid (r<s \wedge s \text{ sudé}) \vee (r>s \wedge s \text{ liché})\}$$
$$Y=\{(r,s)\in P \mid (r<s \wedge s \text{ liché}) \vee (r>s \wedge s \text{ sudé})\}$$
$$Z=\{(r,s)\in P \mid r=s\}$$

(Jedná se o multimnožiny.)

Pro nosič platí $|X|\approx|Y|$, protože je-li $r<s$ pak s může být sudé nebo liché s pravděpodobností 50:50.
Ve skutečnosti je Y o trochu větší než X, ale při dostatečném počtu barev (odstínů) můžeme zanedbat. Pro 4 barvy: # typů párů v X je $\frac{n^2}{2}-n$ a v Y je $n^2/2$, většina náhodných párů v Y má malou pravděpodobnost výskytu. →

X	Y_1	Y_2	Z
(0,2)	(0,3)	(0,1)	(0,0)
(1,2)	(1,3)	(1,0)	(1,1)
(2,1)	(2,0)	(2,3)	(2,2)
(3,1)	(3,0)	(3,2)	(3,3)

Mimochodem pro stegoobjekt je $|Y| > |X|$ resp. $E[|X|-|Y|] < 0$.

Jde o to, že při vkládání se množiny X a Y_1 vyrovnají a množiny Y_2 a Z se vyrovnají.

Konkrétně pro $\alpha = 1$ a $r < s$, jedli $|r-s| > 1$ pak opět pravděpodobnost, že s je sudé vs. liché je 50:50, ale jedli $|r-s| = 1$ pak s je liché.

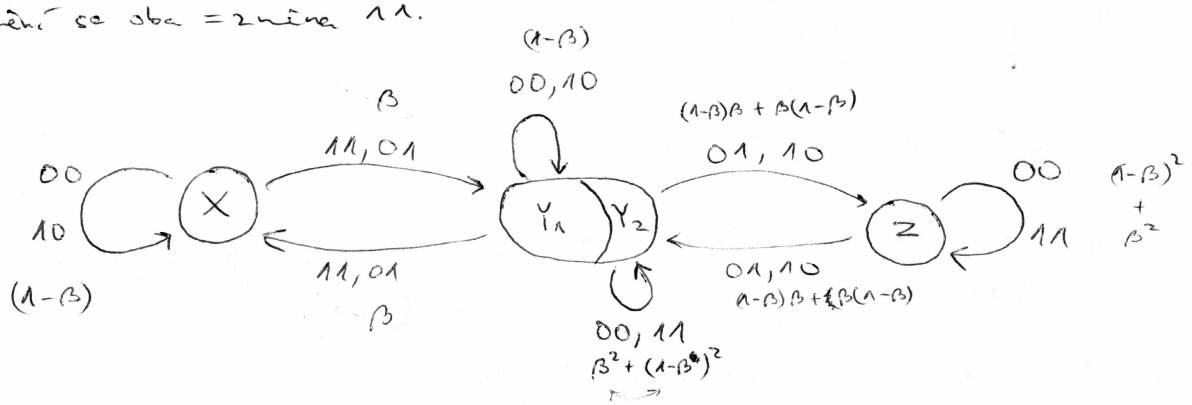
Definujeme $Y_1 := Y \setminus Y_2$

$$Y_2 := \{(r,s) \in P \mid r = 2k,\ s = 2k+1 \text{ nebo } r = 2k+1,\ s = 2k\}$$

Co se může stát s (r,s) při vkládání:

Pozn: Pro Y_2 a Z platí $i=j$
Pro X a Y_1 platí $i \neq j$

1) Nezmění se = změna 00.
2) Změní se jen r = změna 10.
3) Změní se jen s = změna 01.
4) Změní se oba = změna 11.



Označme β pravděpodobnost změny LSB při vkládání, tj:

$\beta := \alpha/2$

dále označme množiny, které vzniknou vkládáním X', Y_1', Y_2', Z'.

$$|X'| \approx (1-\beta)|X| + \beta|Y_1|$$
$$|Y_1'| \approx (1-\beta)|Y_1| + \beta|X|$$
$$|Y_2'| \approx (1-2\beta+2\beta^2)|Y_2| + 2\beta(1-\beta)|Z|$$
$$|X| \approx |Y| = |Y_1| + |Y_2|$$
$$|Y_2| + |Z| = |Y_2'| + |Z'|$$
$$|P| = |X'| + |Y_1'| + |Y_2'| + |Z'|$$

① $|X'| - |Y_1'| = (1-2\beta)(|X| - |Y_1|)$, $|X| - |Y_1| \approx |Y_2|$

② $\Rightarrow |Z| = |Y_2'| + |Z'| - |Y_2|$

③, ④

(Očekávané změny, nežádoucích se chceme zbavit a získat β)

Z těchto rovnic dostaneme

$$2(|Y_2'| + |Z'|)\beta^2 + 2(2|X'| - |P|)\beta + |Y'| - |X'| = 0$$

$$\beta = \frac{1}{2}\left(1 - \frac{|X'|-|Y_1'|}{|Z'|+|Y_2'|} \pm \sqrt{\left(\frac{|X'|-|Y_1'|}{|Z'|+|Y_2'|}\right)^2 + \frac{|Z'|-|Y_2'|}{|Z'|+|Y_2'|}}\right)$$

(α obvykle vycházíš s přesností ±0.05)

Očekáváme $|X'| \geq |Y_1'|$ a $|Z'| \geq |Y_2'|$. V takovém případě je $\beta_- \leq 0.5$ a $\beta_+ \geq 0.5$.
Bereme tedy $\mathrm{Re}(\beta_-)$. (Vkládáme-li hodně dat, tj. β blízko 0,5, může výraz pod odmocninou vyjít záporný. V takovém případě ho zanedbáme tím že vezmeme Re)

Modifikovaný Sample Pairs analysis

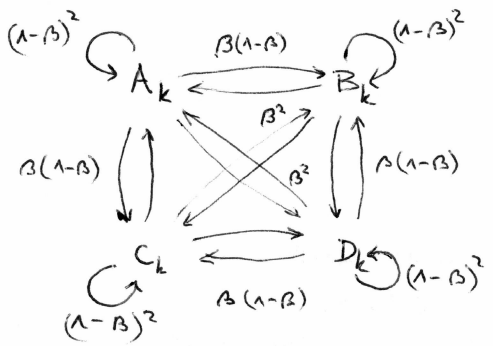
- Označme

$$A_k = \{(2i, 2j) \in P \mid i - j = k\}$$
$$B_k = \{(2i, 2j+1) \in P \mid i - j = k\}$$
$$C_k = \{(2i+1, 2j) \in P \mid i - j = k\}$$
$$D_k = \{(2i+1, 2j+1) \in P \mid i - j = k\}$$

} Multimnožiny

Kde P je multimnožina všech párů sousedících pixelů.

- Co se stane při vkládání:



Označme a_k, b_k, c_k, d_k velikosti multimnožin A_k, B_k, C_k, D_k a a'_k, b'_k, c'_k, d'_k jejich velikosti po vkládání. Potom

$$\begin{pmatrix} a'_k \\ b'_k \\ c'_k \\ d'_k \end{pmatrix} \approx \begin{pmatrix} (1-\beta)^2 & \beta(1-\beta) & \beta(1-\beta) & \beta^2 \\ \beta(1-\beta) & (1-\beta)^2 & \beta^2 & \beta(1-\beta) \\ \beta(1-\beta) & \beta^2 & (1-\beta)^2 & \beta(1-\beta) \\ \beta^2 & \beta(1-\beta) & \beta(1-\beta) & (1-\beta)^2 \end{pmatrix} \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} a_k \\ b_k \\ c_k \\ d_k \end{pmatrix} \approx \frac{1}{1-2\beta} \begin{pmatrix} (1-\beta)^2 & -\beta(1-\beta) & -\beta(1-\beta) & \beta^2 \\ -\beta(1-\beta) & (1-\beta)^2 & \beta^2 & -\beta(1-\beta) \\ -\beta(1-\beta) & \beta^2 & (1-\beta)^2 & -\beta(1-\beta) \\ \beta^2 & -\beta(1-\beta) & -\beta(1-\beta) & (1-\beta)^2 \end{pmatrix} \begin{pmatrix} a'_k \\ b'_k \\ c'_k \\ d'_k \end{pmatrix} \leftarrow$$

- $B_k = \{(2i, 2j+1) \in P \mid i - j = k\}$ vzdálenost: $2i - (2j+1) = 2k - 1$

$C_{k-1} = \{(2i+1, 2j) \in P \mid i - j = k-1\}$ vzdálenost: $2i + 1 - 2j = 2(k-1) + 1 = 2k - 1$

Očekáváme, že v nosiči platí $|B_k| \approx |C_{k-1}|$ protože páry v obou třídách ~~se~~ mají vzdálenost $2k-1$ a liší se pouze paritou. Parita by v nosiči neměla mít vliv na četnost.

- Podíváme-li se na 2. a 3. řádek soustavy dostáváme pro každé k kvadratickou rovnici:

$$\frac{1}{1-2\beta}\left(\beta^2(a'_k + b'_k + c'_k + d'_k) - \beta(a'_k + 2b'_k + d'_k) + b'_k\right) \approx b_k$$

$$\frac{1}{1-2\beta}\left(\beta^2(a'_{k-1} + b'_{k-1} + c'_{k-1} + d'_{k-1}) - \beta(a'_{k-1} + 2c'_{k-1} + d'_{k-1}) + c'_{k-1}\right) \approx c_{k-1}$$

$1-2\beta$ se odstraní a zůstane kvadratická rovnice

Mimochodem $|A_k| \approx |D_k|$ se využít nedá, protože to implikuje že $|A'_k| \approx |D'_k|$ VB.

Jak naložit s touto sadou kvadratických rovnic?

1) Pro např. $i = -2, -1, 0, 1, 2$ najdu kořeny kvadratické rovnice a z nich vybere minimální.

2) Kvadratické rovnice sečtu pro všechna i sudá nebo pro i lichá, dostanu jednu rovnici a tu vyřeším.
(Kdybych sečetl přes všechna i tak se mi průběžně pozrušují některé členy a ve výsledku dostanu lineární rovnici jejímž řešením je $\beta = 0.5$.)

3) Nebudu vůbec řešit kvadratické rovnice tj. nebudu předpokládat $b_k \doteq c_{k-1}$ ale zvolím β tak, aby b_k bylo co nejblíže c_{k-1} pro všechna k.

$$\hat{\beta} = \arg\min_{\beta} \sum_{k} (b_k - c_{k-1})^2$$

kde za b_k a c_{k-1} dosadíme výrazy obrahující čárkovaná písmena.

Pokus s $\beta = 0.2$: Průměr odchylky $|\beta - \hat{\beta}|$ pro různé metody:

SPA: 0.0077 (1): 0.0466 (2) $k = -50 \ldots 50$ sudé: 0.0101 (3) $k = -50, \ldots 50$: 0.0056

Vkládání do paletových formátů

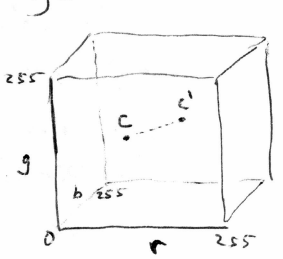
1) Změna uspořádání palety
+ není vůbec znát od pohledu na obrázek
- omezená kapacita $\log_2(256!) \approx 1684$ bitů ≈ 210 bajtů
- většina programů uspořádá paletu podle jasu nebo frekvence výskytu dané barvy v obrázku $\Rightarrow$ náhodně uspořádaná paleta je podezřelá.
- otevření a opětovné uložení obrázku může vést k setřídění palety a ke ztrátě zprávy.

2) Vkládání do matice ukazatelů

a) Paletu setřídíme podle jasu (a přeindexujeme matici ukazatelů). Potom provedeme LSB embedding na ukazatelích. (EzStego)
- Problém: změna nejnižšího bitu sice neovlivní žádným způsobem jas pixelu ale může vést ke změně barvy např. červené na zelenou.

b) zvolíme sofistykovanější uspořádání palety, které minimalizuje vzdálenost mezi následujícími barvami v paletě.
(následujícími po sobě)

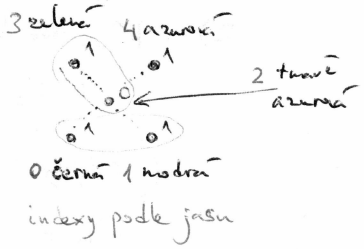
- Problém: NP-úplný (obchodní cestující).



c) Nebudeme vkládat do LSB

Ke každé barvě v paletě přiřadíme hodnotu 0 nebo 1 takovým způsobem, aby nejbližší barva v paletě měla opačnou hodnotu.

Vkládání s optimální volbou parity



Pozn: Přiřazení parity nemusí být vyvážené.

- Algoritmus optimální volby parity

Vstup: Množina $\{c_i : i \in I\} \subseteq \{0,1,\dots,255\}^3$ (paleta). $I = \{0,1,\dots,255\}$ množina indexů.
bez duplicit

Výstup: $p : I \to \{0,1\}$,
$f : I \to I$ takové, že
$p(i) = 1 - p(f(i)) \quad \forall i \in I$
$\|c_i - c_{f(i)}\|_2$ nabývá minimální možnou hodnotu

Postup:

```
for each i∈I
    p(i) := ∞
E := {(||c_i − c_j||_2, c_i, c_j) : i,j∈I, i≠j}
#Množinu E lze uspořádat lexikograficky, tj. především podle
#vzdáleností barev a potom podle barev jakožto 3-složkových vektorů.
while E ≠ ∅
    (d, c_i, c_j) := min_LEX E   #p(i)=∞. Jakmile p(i)∈{0,1}, barva c_i se odstraní
                                 #ze všech výskytů na druhé pozici.
    if p(j) = ∞                  #jestliže p(j)=∞ jinak p(i)=∞
        p(i) := 0
        p(j) := 1
        f(i) := j
        f(j) := i
        E := E \ (ℝ × {c_i, c_j} × C)   odstraní vše tvaru (·, c_i, ·) a (·, c_j, ·)
    else
        p(i) := 1 − p(j)
        f(i) := j
        E := E \ (ℝ × {c_i} × C)        odstraní vše tvaru (·, c_i, ·)
    end if
done
```

- Algoritmus je deterministický a závisí jen na množině barev v paletě (nikoliv na jejím uspořádání)
- Doběhne: množina E se při každém průchodu zmenší.
- Funkce p i f jsou dobře definované
  - Zřejmě platí $p(i) = 1 - p(f(i)) \quad \forall i \in I$
  - $\|c_i - c_{f(i)}\|_2$ nabývá min možnou hodnotu.

$\left(p(i) = \infty \Leftrightarrow (\cdot, c_i, \cdot) \in E \right)$

- Algoritmus vkládání s optimální volbou parity

Vstup: Nosič $x=(x_1,\dots,x_n)\in I^n$ posloupnost indexů do palety.

zpráva $z=(z_1,\dots,z_m)\in\{0,1\}^m$

klíč $\pi\in S_n$

Výstup: Stegoobjekt $y=(y_1,\dots,y_n)\in I^n$

Postup: for each $i\in\{1,2,\dots,m\}$

$$y_{\pi(i)}=\begin{cases} x_{\pi(i)} & \text{pokud } p(x_{\pi(i)})=z_i \\ f(x_{\pi(i)}) & \text{pokud } p(x_{\pi(i)})\neq z_i \end{cases}$$

for each $i\in\{m+1,\dots,n\}$

$$y_{\pi(i)}=x_{\pi(i)}$$

- Algoritmus extrakce

Vstup: Stegoobjekt $y=(y_1,\dots,y_n)\in I^n$

Klíč $\pi\in S_n$

Délka zprávy m

Výstup: Zpráva $z=(z_1,\dots,z_m)\in\{0,1\}^m$

Postup: for each $i\in\{1,\dots,m\}$

$$z_i=p(y_{\pi(i)})$$

- Efektivita vkládání: $e=\dfrac{E[\log_2|M(x)|]}{E[d_2(x,y)]}=\dfrac{m}{E[d_2(x,y)]}$

definujme $h(i):=\#\{j\mid x_j=i\}$

$$E[d_2(x,y)]=\sum_{j=1}^{n}E[\|c_{x_j}-c_{y_j}\|^2]=\sum_{j=1}^{n}\underbrace{\frac{1}{2}\frac{m}{n}}_{\substack{\text{pravděpodobnost, že}\\ \text{j-tý pixel bude}\\ \text{změněn}}}\underbrace{\sum_{i\in I}\frac{h(i)}{n}\|c_i-c_{f(i)}\|^2}_{\substack{\text{očekávaná velikost}\\ \text{změny}}}$$

$$=\frac{m}{2}\sum_{i\in I}\underbrace{\frac{h(i)}{n}}_{\substack{\text{relativní četnost}\\ \text{barvy } i \text{ v obrázku}}}\|c_i-c_{f(i)}\|^2 \leftarrow \text{vzdálenost k nejbližší barvě}$$

Efektivita $e=\dfrac{2}{\sum_{i\in I}\frac{h(i)}{n}\|c_i-c_{f(i)}\|^2}$

efektivita	# barev	
barvy.gif	255	23
optpar	0.0228	0.0012
lumLSB	0.0019	0.0003
	12×	4×

LSB 10

Vkládání při redukci hloubky barev s optimální volbou parity

Vezmeme jako nosič 24-bitový rastrový obrázek a provedeme redukci hloubky barev s difuzí chyby, ovšem s tím rozdílem, že při zaokrouhlování na nejbližší barvu se u pixelů nesoucích bit zprávy omezíme na množinu barev jejichž parita odpovídá bitu zprávy.

Adaptivní vkládání

- Nechceme vkládat bity zprávy do oblastí uniformní barvy jako např. nebe nebo přesvícené oblasti. Chceme vkládat do oblastí s vysokou texturou.
- Problém: Příjemce nemá k dispozici nosič, aby určil co je oblast s vysokou či nízkou texturou. (vkládání může texturu snížit)
- Příklad: -Obrázek rozdělíme na bloky velikosti 3×3 a definujeme funkci t, která určí texturu bloku, např.

  $t(B) = \#$ barev v B.

  -Do daného bloku vkládáme právě když jeho textura $t(B) > p$, kde p je nějaká vhodně zvolená prahová hodnota.
- Poté co provedeme vkládání do bloku, musíme texturu změřit znovu. Stane-li se, že textura klesne pod prahovou hodnotu, bude tento blok příjemcem ignorován. Bity proto musíme vložit znovu do následujícího bloku. Tímto se snižuje efektivita vkládání.

Barn23.gif: $\gamma = 2$ ... 53% pixelů použito
$\gamma = 3$ ... 26% pixelů použito

Párová analýza paletových formátů

LSB 11

Kvantitativní útok na Ez Stego

Ez Stego:
Máme paletový obrázek, paleta je setříděna podle jasu.

Nosič: $x \in \{0,1,\dots,255\}^{n}$ indexy do palety $\beta = 1/2$ pravděpodobnost změny

Stegoobjekt $y \in \{0,1,\dots,255\}^{n}$

Zprávu vkládáme do LSB, čili opět $\dots\ 2k,\ 2k+1,\ 2k+2,\ 2k+3,\dots$

Očekáváme, že v přirozeném nosiči se objeví posloupnosti stejné barvy. Tato vlastnost je vkládáním narušena.

Definice

k-tý barevný výřez $\varphi_k(c) := \begin{cases} 0 & \text{pro } c = 2k \\ 1 & \text{pro } c = 2k+1 \\ \varepsilon & \text{pro ostatní } c \end{cases}$ ↑ prázdné slovo.

k-tý posunutý barevný výřez $\varphi'_k(c) := \begin{cases} 0 & \text{pro } c = 2k-1 \\ 1 & \text{pro } c = 2k \\ \varepsilon & \text{pro ostatní } c \end{cases}$

přirozeným způsobem rozšíříme na slova (posloupnosti) jako homomorfismus monoidů.

$Z := \varphi_0(y) \,\|\, \varphi_1(y) \,\|\, \dots \,\|\, \varphi_{127}(y)$

$Z' := \varphi'_1(y) \,\|\, \varphi'_2(y) \,\|\, \dots \,\|\, \varphi_{127}(y)$

$R(\beta) = \dfrac{\#\{i \mid z_i = z_{i+1},\ 1 \le i \le n-1\}}{n-1}$ Relativní počet homogenních párů v Z.

$R'(\beta)$ obdobně

Př

$y = 0\,1\,1\,2\,3\,1\,2\,2$ (φ_0: ↑ ↑ ↑ ε; φ_1: ↓ ↓ ↷ ε)

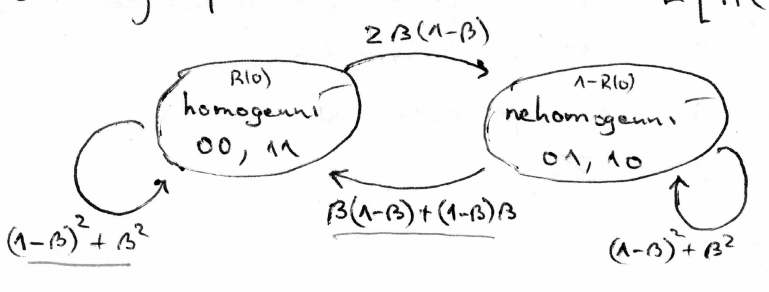
$\rightsquigarrow Z = 0\,1\,1\,1\,0\,1\,0\,0$ $R(\beta) = \frac{3}{7}$

$0\,1\,1\,2\,3\,1\,2\,2$ (φ'_1: ↓ ↓ ↓ ↷ ↷)

$\rightsquigarrow Z' = 0\,0\,1\,0\,1\,1$ $R'(\beta) = \frac{2}{5}$

- V nosiči očekáváme, že počty homogenních párů v neposunutých a posunutých výřezech budou stejné: $\boxed{R(0) = R'(0)}$ $\left(E[R(0) - R'(0)] = 0\right)$
- R je kvadratická funkce:

co se děje při vkládání:



$E[R(\beta)] = R(0)\left((1-\beta)^2+\beta^2\right) + \left(1-R(0)\right) 2\beta(1-\beta)$

$R(0)$ je konstanta pro pevně zvolený nosič a $R(\beta)$ je tedy kvadratický polynom v proměnné β.

$\boxed{E[R(\beta)] = E[R(1-\beta)]}$

- **$R'(1-\beta)$** zjistíme tak, že ve stegoobjektu y obrátíme všechny LSB, spočítáme posunuté výřezy a určíme počet homogenních párů.

Proč to funguje:

máme $y = 0\,1\,\boxed{1}\,2\,\boxed{3}\,\boxed{1}\,2\,2$, dejme tomu, že $\beta = \frac{3}{8}$ a

y vzniklo překlopením tří LSB, takže

$$x = \underline{0}\,\underline{1}\,0\,\underline{2}\,\underline{2}\,0\,\underline{2}\,\underline{2}$$

nyní simulujme co by se stalo kdyby $\beta = \frac{5}{8}$

$$\bar{y} = 1\,0\,0\,3\,2\,0\,3\,3 \qquad R'(1-\beta) = \frac{0}{1} = 0$$

Nezrušíme sice β ani pozice, kde došlo k překlopení, ale překlopením všech hodnot simulujeme vkládání s parametrem $1-\beta$.

- **$R(1/2) \approx 1/2$**

Pro $\beta = 1/2$ tvoří LSB náhodnou posloupnost.
Z je vlastně posloupnost LSB akorát trochu přepermutovaná.
$P(z_i = z_{i+1}) = 1/2$ pro libovolné i, $1 \le i \le n-1$

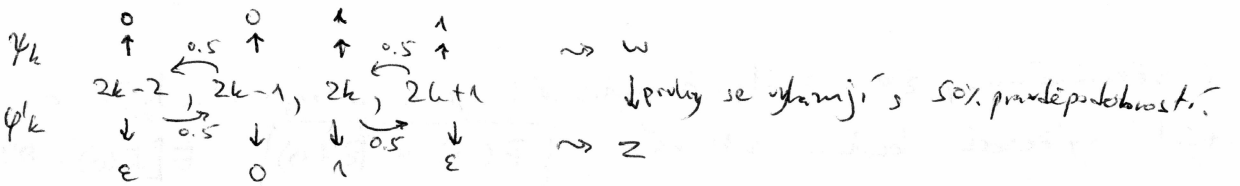
$R(1/2) = \frac{1/2\,(n-1)}{n-1}$ ← očekávaný počet homogenních párů.

- $R'(\frac{1}{2}) = ?$

- Definujeme: $\psi_k(c) \begin{cases} 0 & \text{pro } c \in \{2k-2, 2k-1\} \\ 1 & \text{pro } c \in \{2k, 2k+1\} \\ \varepsilon & \text{pro ostatní } c \end{cases}$ Pozorování: $\psi_k(x) = \psi_k(y)$

$$W := \psi_1(y) \,\|\, \psi_2(y) \,\|\, \cdots \,\|\, \psi_{127}(y)$$

- Je-li $\beta = 1/2$, pak Z vznikne z W tak, že každý člen posloupnosti W buď vyhodíme nebo zachováme s pstí 50%.



- Ať $W = \ldots w_i, \ldots, w_{i+k} \ldots$ jestliže $w_i = w_{i+k}$, pak pravděpodobnost, že v Z utvoří homogenní pár je $(\frac{1}{2})^{k+1}$ tj. $1/2$ že w_i se zachová, $1/2$ že w_{i+k} se zachová a $(1/2)^{k-1}$ že ty mezi nimi vypadnou.
- $\sum_{k=1}^{n-1} \frac{1}{2^{k+1}} \#\{i \mid w_i = w_{i+k},\ 1 \le i \le n-k\}$ kde n je délka W.
očekávaný počet homogenních párů v Z
- Očekávaná délka Z je $n/2$.

$$R'(1/2) = \frac{1}{\frac{n_w}{2}-1} \sum_{k=1}^{n_w-1} \frac{1}{2^{k+1}} \#\{ i \mid w_i = w_{i+k},\ 1 \le i \le n_w - k \}$$

kde n_w je délka w.

- Praxe ukazuje, že $R'(\beta)$ lze dobře modelovat kvadratickým polynomem. (Důkaz ovšem neznáme.)

Předpokládejme tedy, že $f(\beta) := R(\beta) - R'(\beta)$ je kvadratický.

Známe $f(0) = R(0) - R'(0) \approx 0$.

$f(1/2) = R(1/2) - R'(1/2) \approx 1/2 - R'(1/2)$

$f(\beta) = R(\beta) - R'(\beta)$

$f(1-\beta) = R(1-\beta) - R'(1-\beta) = R(\beta) - R'(1-\beta)$

Polynom známe ve čtyřech bodech $\Rightarrow$ redundance? Nikoliv, některé body jsou totiž závislé na neznámé β.

. . . technický postup . . .

Odtud dostaneme kvadratickou rovnici $a\beta^2 + b\beta + c = 0$, kde

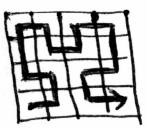
$a = 2 - 4R'(1/2)$

$b = R'(\beta) - R'(1-\beta) - 2 + 4R'(1/2)$

$c = R(\beta) - R'(\beta)$

↑ (pozor, toto není f)

Spočítáme $\hat{\beta} = \min\left(\mathrm{Re}(\beta_+), \mathrm{Re}(\beta_-)\right)$. (Pro $\beta \approx 0.5$ můžou vyjít komplexní.)

- Pozn obrázek můžeme načíst buď  anebo po Hilbertově křivce:  Tímto se zvyšuje pravděpodobnost, že zachytíme uniformní oblast obrázku v posloupnosti
- Pozn Odhad opět vychází ± 0.05

Útok na vkládání s optimálním přiřazením parity

OPTPAR ATTACK 1

Idea:
- 2 histogramy stegoobrázku zrekonstruujeme histogram nosiče pro různé hodnoty α.
- Jestliže α zvolíme větší než je jeho skutečná hodnota objeví se v zrekonstruovaném histogramu zapamatované hodnoty.
- Předpokládáme, že barvy v histogramu nenají rovnoměrné rozdělení.

Př:

Vkládání:

$h(1) \doteq (1-\beta)\, h_0(1)$

$h(2) \doteq (1-\beta)\, h_0(2) + \beta \cdot h_0(1)$

$h(3) \doteq (1-\beta)\, h_0(3)$

$h(4) \doteq (1-\beta)\, h_0(4) + \beta (h_0(2) + h_0(3))$

$h_0(1) = 1000$

$h_0(2) = 10$

$\alpha = 0{,}2$

$\beta = 0{,}1 = \alpha/2$ (pst. změny)

$\Downarrow$

$h(1) \approx 900$

$h(2) \approx 9 + 100 = 109$

Rekonstrukce:

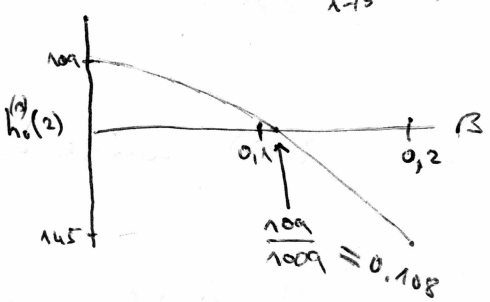
$h_0^{(\beta)}(1) \doteq \frac{1}{1-\beta}\, h(1)$

$h_0^{(\beta)}(2) \doteq \frac{1}{1-\beta}(h(2) - \beta \cdot h_0(1))$

$\doteq \frac{1}{1-\beta}\left(h(2) - \frac{\beta}{1-\beta} h(1)\right)$ ←

$h_0^{(\beta)}(3) \doteq \dots$

$h_0^{(\beta)}(2) \approx \frac{1}{1-\beta}\left(109 - \frac{\beta}{1-\beta} 900\right)$

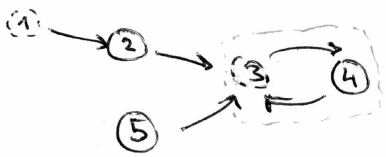


Změny v histogramu při vkládání lze popsat maticově:

Definujeme $A = (a_{ij}) \in \mathbb{R}^{|C| \times |C|}$

$$a_{ij} = \begin{cases} 1 & \text{pokud } f(j) = i \\ 0 & \text{jinak} \end{cases}$$

Př:



$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Vkládání

$$\vec{h} = \begin{pmatrix} h(1) \\ \vdots \\ h(|C|) \end{pmatrix} \quad \vec{h}_0 = \begin{pmatrix} h_0(1) \\ \vdots \\ h_0(|C|) \end{pmatrix},$$

$$\vec{h} \approx (\beta \cdot A + (1-\beta) I)\, \vec{h}_0$$

Rekonstrukce

$$\vec{h}_0^{(\beta)} = (\beta \cdot A + (1-\beta) I)^{-1} \vec{h}$$

Útok

$\forall \beta \in [0,1]$ zrekonstruujeme histogram h_0 a definujeme

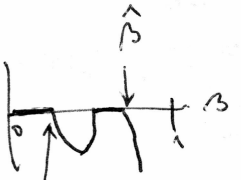
$$y(\beta) = \min_i h_0^{(\beta)}(i)$$

Za $\hat{\beta}$ zvolíme ~~kořen~~ maximální hodnotu pro kterou $y(\beta)$ mění znaménko z + na −.
v intervalu ~~[0,1]~~ $[0, 1/2]$

Potom $\hat{\beta}$ je horní odhad pravděpodobnosti změny.

$(\hat{\alpha} = 2\hat{\beta})$.

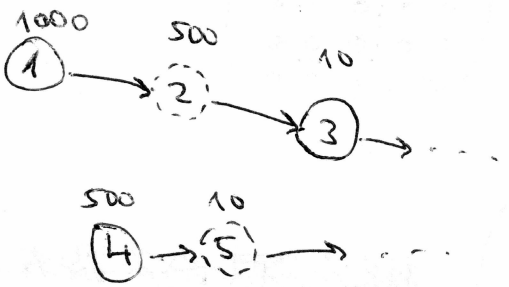
Pozn $y(\beta)$ může mít tvar



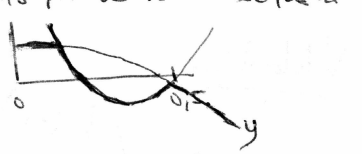
Alternativní volba $\hat{\beta}$.

(Naše volba $\hat{\beta}$ jako maximálního kořenem je v souladu s myšlenkou, že se jedná o horní odhad β.)

Smyslem volby maximální hodnoty je odignorovat případnou parabolu, protože ta se chová divně.
Viz Mathematica:



Toto není důvod proč to pro velké α selhává



Funguje dobře pro $\alpha < 0.5$

Když vypustíme ~~jáčkovy~~ bary v jádrech, pak funguje dobře pro $\alpha > 0.5$